Ear segmentation using Mask R-CNN and Cascade R-CNN

Assignment #2

Image Based Biometrics 2021/22, Faculty of Computer and Information Science, University of Ljubljana

Novak Marko

***Abstract*—Mask R-CNN and Detectron2 are currently state-of-the-art solutions for one-step object detection and segmentation. This report applies the approach to ear segmentation on the AWE database [?].**

I. Introduction

Faster R-CNN [1] was proposed in 2015 as a solution offering simultaneous detection and classification of objects. In 2017 an upgrade called Mask R-CNN [2] was proposed, which introduced the ability to precisely segment the target objects using pixel masks. Later the same year, Cascade R-CNN [3] was proposed, improving detection on smaller feature pyramid samples and overall model robustness on different sizes of inputs.

This report is based on Tensorpack [4] implementation of Mask R-CNN.

II. Methodology

AWE dataset comes with bounding boxes and ear masks. Each mask is split according to bounding boxes for training to get an array of one mask per instance. These masks are stored as NumPy files on start and reused on each run for faster training.

The Mask R-CNN variant chosen for this task uses ResNet101 backbone CNN, a feature pyramid network head, and a cascade wrapper around the head network. Since the problem at hand deals with the segmentation of relatively small objects (ears) on larger images, cascade should provide some improvement over plain Mask R-CNN architecture with feature pyramids. Both backbone and head networks use group normalisation, which is a normalisation technique that doesn't depend on batch size. That improves detection, as regular batch normalisation varies a lot for small batches with an inconsistent number of target objects.

The model has been trained using the training dataset containing 750 images. It's been trained over 144 epochs, totaling 72 000 steps (500 steps per epoch). Training images from AWE dataset have been randomly flipped and cropped to increase the sample size for training. Initial weights for the model have been loaded from a model pre-trained on COCO dataset.

The final training took about 15 hours on a single P100 GPU.

III. Results

The resulting model has been tested on 250 test (validation) images from AWE dataset. The model was able to score true matches with high confidence, making it possible to achieve 99,3% true positive rate at just 11 false positives (Figure 1).

True positive rate in realtion to false positives gives us some general idea about the performance of the model, but in order to better understand the performance of the model, additional metrics are neccesary. The following metrics were calculated based on segmentation mask:

$$IoU = area(m_p \cap m_{gt})/area(m_p \cup m_{gt})$$

$$precision = area(m_p \cap m_{gt})/area(m_p)$$

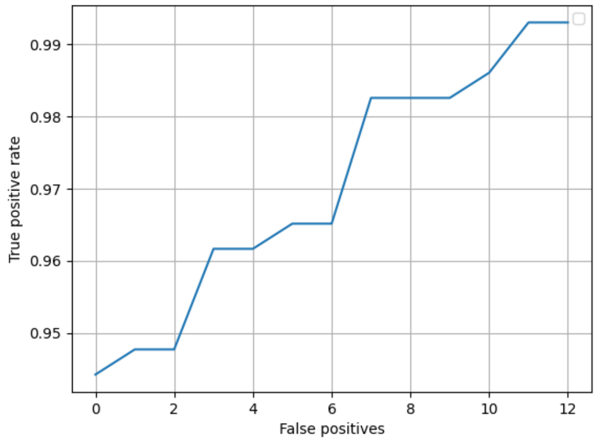

Figure 1: ROC curve for the resulting model.

$$recall = area(m_p \cap m_{gt})/area(m_{gt})$$

Since false positives have small or no intersection between m_p (predicted mask) and m_{gt} (ground truth mask), it doesn't make sense to look at metric values relative to false positives. Instead, a confidence score from the output has been used, and a false positive count has been shown as an additional metric on the plot (figure 2).

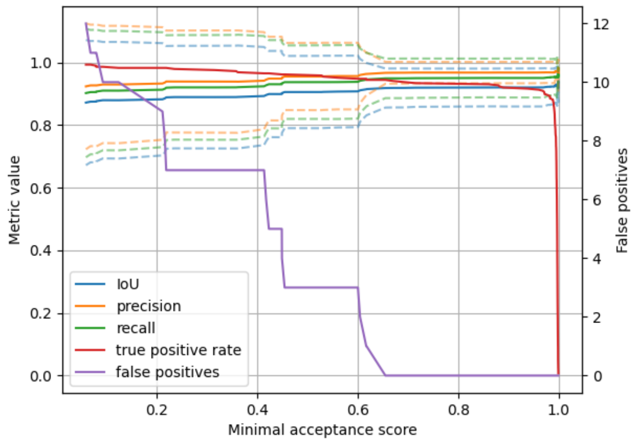

Figure 2: Metrics relative to minimal acceptance score.

The minimal acceptance score threshold of 0,65 is high enough to eliminate all false positive matches and obtain results as shown in table I. Using a higher threshold such as 0,90 only minimally improves the metrics but decreases true positive rate by almost 3%, so a larger test dataset would be needed to better determine what a good threshold value might be.

Table I: Metric values at *score* $> 0{,}65$ and *score* $> 0{,}90$

TPR	IoU	precision	recall
94,43%	91,85% ±6,22%	96,67% ±3,50%	94,91% ±6,30%
91,64%	92,02% ±6,08%	96,77% ±3,41%	95,03% ±6,24%

IV. Conclusion

This report seems to indicate some improvement for ear detection and segmentation using Mask R-CNN over research presented in Mask R-CNN for Ear Detection [5], which is likely due to the use of cascade wrapper and better normalisation methods. However, while both models use the same training and validation datasets, I was unable to find the exact code used to calculate the metrics in the original paper. Because of that, raw predictions data is included with the report for further analysis.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[3] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," 2017.

[4] Y. Wu *et al.*, "Tensorpack," https://github.com/tensorpack/, 2016.

[5] M. Bizjak, P. Peer, and Z. Emeršič, "Mask r-cnn for ear detection," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 1624–1628.